Releasing reliable, secure code in today's market is nearly impossible due to the demand for increasingly sophisticated software coupled with shorter product cycles. In addition, much of today's software is expected to provide five 9s of reliability while sitting on a network in an untrusted environment. The result is a daunting IT burden that is making the cost of developing and supporting software higher than ever before.

Coverity's solutions dramatically reduce the cost of software development and testing using technology that represents a huge leap forward in automated software analysis. Coverity's patent pending source code analysis technology was originally developed by a team of researchers in the Computer Systems Lab at Stanford University. Initial applications of the technology resulted in the successful detection of over 2000 defects and hundreds of exploitable security holes in Linux.

Software Analysis Toolset

Coverity, Inc.'s Software Analysis Toolset (SWAT) automatically pinpoints the sources of costly software defects and security holes before the code ever leaves the developer's desktop. The traditional software development cycle is enhanced by an additional phase: the automatic code audit. The toolset provides:

- Powerful, push-button detection of common software errors.
- Scalability to millions of lines of code within a small fraction of the build time.
- An easy to use, extensible architecture that allows customers to translate company-specific defects found during traditional QA into automated analyses.
- Automatic customization to each client's source code through patent pending statistical inference technology.

Coverity's products find defects that are costly or even impossible to detect through the traditional QA process. In addition, the extensible architecture of SWAT easily turns each error detected by the normal QA process into a list of similar errors in other parts of the source code by allowing for quick and easy development of custom analyses.

Coverity's toolset provides significant enhancements to traditional QA:

- Defects are automatically detected as soon as the developer is ready to compile the code.
- Error diagnosis is fast, easy to use, and precise.
- Every path through the code is examined.
- No annotations or test-cases are needed.

In addition, Coverity's web-based user interface maintains comprehensive reports of defect detection and repair, providing instant identification of hot spots in the code and comprehensive code hardening metrics.

Running SWAT

The Software Analysis Toolset runs on each developer's desktop. Before each developer checks any code into the main repository, the code must pass an audit with SWAT. Coverity also provides more in depth analysis solutions that can run as a companion to a nightly build. Bugs detected by the toolset are stored in a bug database that allows for easy inspection through our simple, web-based user interface that integrates seamlessly with most bug tracking systems.

For example, SWAT's memory leak detector describes each detected leak step by step through the web-based interface:

```
==>
LEAK detector: leaking storage vc
File: linux/drivers/media/dvb/av7110/saa7146_v4l.c:58
Function: saa7146_v4lclip2plain
Line Number: 58
(also see line 65)

58          vc = vmalloc(sizeof(struct video_clip)*(clipcount));
59          if( NULL == vc) {
60                  printk("saa7146_v4l.o: ==> v4lclip2saa7146_v4l.o: no memory #2!\n");
61                  return -ENOMEM;
62          }
63          if(copy_from_user(vc,clips,sizeof(struct video_clip)*clipcount)) {
64                  printk("saa7146_v4l.o: ==> v4lclip2saa7146_v4l.o: could not copy
from user-space!\n");
==>
LEAK detector: Missing deallocation of pointer vc.
Line Number: 65
(also see line 58)
65                  return -EFAULT;
66          }
```

In addition to memory leaks, SWAT automatically detects a broad range of defects including the following general categories:

- Memory corruption*
- Array/buffer overrun
- NULL pointer dereference*
- API checking*
- Concurrency*
- Security
- Cross interface error handling*
- Useless operations/dead code
- 64-bit compliance
- Coding standards

* categories that include patent pending statistical inference technology

Coverity: Under the Hood

The basics of Coverity's technology are described in detail in the following references:

- "Checking System Rules Using System-Specific, Programmer-Written Compiler Extensions." *Usenix Symposium on Operating Systems Design and Implementation, October 2000.*
    - Describes the basics of Coverity's technology. Won best paper award at OSDI 2000.
    - http://www.stanford.edu/~engler/mc-osdi.pdf

- "Using Programmer-Written Compiler Extensions to Catch Security Holes." *IEEE Symposium on Security and Privacy, May 2002.*
    - Describes early successes in applying Coverity's technology to catch one class of security holes that are prevalent in operating systems and other low-level software.
    - http://www.stanford.edu/~engler/sp-ieee-02.pdf

- "Bugs as Deviant Behavior: A General Approach to Inferring Errors in Systems Code." *ACM Symposium on Operating Systems Principles, October 2001.*
    - Provides a technical description of Coverity's statistical inference capabilities, which automatically adapt the set of rules we check to each new code base.
    - http://www.stanford.edu/~engler/deviant-sosp-01.pdf

- "A System and Language for Building System-Specific, Static Analyses." *ACM SIGPLAN Conference on Programming Language Design and Implementation, June 2002.*
    - Provides the most up-to-date technical description of Coverity's analysis engine.
    - http://www.stanford.edu/~engler/p27-hallem.pdf

About Coverity

Coverity, Inc. is a leading provider of source code analysis solutions that help organizations produce reliable, secure software while significantly improving time to market. Coverity's quickly growing customer base includes a wide range of companies, from startups to Fortune 100 enterprises.

Coverity's patent pending technology was originally developed by a team of researchers in the Computer Systems Lab at Stanford University. Preliminary applications of the technology resulted in the successful detection of over 2000 defects and hundreds of exploitable security holes in the Linux and OpenBSD kernels.

Coverity, Inc. is headquartered in Menlo Park, California. For more information or a free trial, contact us at:

Tel: 1-650-980-3408
E-mail: info@coverity.com

